Writing for Computer Science
Justin Zobel, 2ed.,
Springer
for CS
601 only

10 Doing research

The intensity of the conviction that a hypothesis is true has no bearing on whether it is true or not.

P. B. Medawar
Advice to a Young Scientist

The great tragedy of Science, the slaying of a beautiful hypothesis by an ugly fact.

T. H. Huxley
Biogenesis and Abiogenesis

An argument is a connected series of statements intended to establish a proposition . . . . Argument is an intellectual process. Contradiction is just the automatic gainsaying of anything the other person says.

Monty Python
The Argument Sketch

A research paper, written up and submitted for refereeing, is the result of a process of research that may have been proceeding for months or years. It is not an end-product, but more typically describes recent results or is a preliminary study. It is rare that a write-up is final, concluding forever a program of research on a topic; however, the write-up is based on a great deal of activity. Indeed, with just a few pages representing months or more of work by several people, a paper may be only a tiny window into the research.

A paper, then, is an outcome of a cycle of activity, from speculation through definition and experimentation to write-up, with a range of obstacles and issues that can arise on the way. In this chapter I review the process of research, in particular the early stages of a project. The perspective taken is from the ground, as a working scientist: What kinds of stages and events does a researcher have to manage in order to produce an interesting, valid piece of research? This chapter and Chapter 11 complement the preceding parts of the book—on the topic of how research should be described—by considering how the content of a paper is arrived at.

Philosophers and historians of science have reflected at length on the meaning, elements, and methods of research, from both practical and abstract points of view. These reflections can be of great benefit to a working scientist. Any competent researcher can learn from an alternative perspective on their work, and being able to describe what we do helps us to understand whether we are doing it well.

At the same time, learning to do research involves piecemeal acquisition of a range of specific skills. Only with experience does a student see these skills as part of a single integrated "process of research". That is, many people learn to be scientists by doing research stage-by-stage under supervision, and only after having been through the research process does the bigger picture become evident. For that reason, for novices the correspondence between abstractions of research and a particular investigation can be hard to identify.

A related problem is that newcomers to research may initially draw inappropriate analogies to activities with which they are already familiar. For example, in computer science many research students see experimentation as a form of software development, and undertake a research write-up as if they were assembling a user manual or software documentation. Part of learning to be a computer scientist is recognizing how the aims of research differ from those of coursework or programming.

Beginnings

The origin of a research investigation is typically a moment of insight. A student attending a lecture wonders why search engines do not provide better spelling correction. A researcher investigating external sorting is at a seminar on file compression, and ponders whether one could be of benefit to the other. A user is frustrated by network delays and questions whether the routing algorithm is working well. A student asks a professor about the possibility of research on evaluation of code functionality; the professor, who hadn't previ-

ously contemplated such work, realises that it could build on recent advances in type theory.

Research ideas often come to mind when the brain is idling, or when separate topics coincidentally arise at the same time. Tea-room arguments are a rich source of seed ideas. One person is idly speculating, just to make conversation; another pursues the speculation and a research topic is created. Or someone claims that a researcher's idea is unworkable, and a listener starts to turn over the arguments. What makes it unworkable? How might those issues be addressed?

This first step is a subjective one: to choose to explore ideas that seem likely to succeed, or are intriguing, or have the potential to lead to something new, or contradict received wisdom. At this stage, it isn't possible to know whether the work can lead to valuable results; otherwise there would be no scope for research. The final outcome is an objective scientific report, but curiosity and guesswork are what establish research directions.

It is typically at this stage that a student becomes involved in the research. Some students have a clear idea of what they want to pursue—whether it is feasible, rational, or has research potential is another matter—but the majority are in effect shopping for a topic and advisor. They have a desire to work on research and to be creative, perhaps without any definite idea of what research is. They are drawn by a particular area or problem, or want to work with a particular individual. Students may talk through a range of possible projects with several alternative advisors before making a definite choice and starting to work on a research problem in earnest.

Shaping a research project

How a potential research topic is shaped into a concrete project depends on context. Experienced scientists aiming to write a paper on a subject of mutual interest tend to be fairly focused: they quickly design a series of experiments or theoretical goals, investigate the relevant literature, and set deadlines.

For students, undertaking research involves training, which affects how the work proceeds. Also, for a larger research program such as a PhD, there are both short-term and long-term goals: the current specific explorations, which may be intended to lead to a research paper, and their role as a part of a wider investigation that will eventually form the basis of the student's thesis.

At the beginning of a research program, then, you need to establish answers to two key questions. First, what is the broad problem to be investigated? Second, what are the specific initial activities to undertake and outcomes to pur-

sue? Having clear short-term research goals gives shape to a research program. It also gives the student training in the elements of research: planning, reading, programming, testing, analysis, critical thinking, writing, and presentation.

For example, in research in the early 1990s into algorithms for information retrieval, we observed that the time to retrieve documents from a repository could be reduced if they were first compressed; the cost of decompression after retrieval was outweighed by savings in transfer times. A broad research problem suggested by this topic is whether compression can be of benefit within a database even if the data is stored uncompressed. Pursuing this problem with a research student led to a specific initial research goal (used as a running example in Chapter 9): given a large relation that is compressed as it is read into memory, is it possible to sort it more rapidly than if it were not compressed at all? What kinds of compression algorithm are suitable? Success in these specific explorations leads to questions such as, where else in a database system can compression be used? Failure leads to questions such as, under what conditions might compression be useful?

When developing a question into a research topic, it is helpful to explore what makes the question interesting. Productive research is often driven by a strong motivating example, which also helps focus the activity towards useful goals. It is easy to explore problems that are entirely hypothetical, but difficult to evaluate the effectiveness of any solutions. Sometimes it is necessary to make a conscious decision to explore questions where work can be done, rather than where we would like to work; just as medical studies may involve molecular simulations rather than real patients, robotics may involve the artifice of soccer-playing rather than the reality of planetary exploration.

In choosing a topic and advisor, many students focus on the question of "is this the most interesting topic on offer?", often to the exclusion of other questions that are equally important. One such question is "is this advisor right for me?" Students and advisors form close working relationships that, in the case of a PhD, must endure for several years. The student is typically responsible for most of the effort, but the intellectual input is shared, and the relationship can grow over time to be a partnership of equals. However, most relationships have moments that are less than harmonious. Choosing the right person—considering the advisor as an individual, not just a respected researcher—is as important as choosing the right topic. A charismatic or famous advisor isn't necessarily likeable or easy to work with.

The fact that a topic is in a fashionable area should be at most a minor consideration; the fashion may well have passed before the student has graduated. Some trends are profound shifts that have ongoing effects, such as the oppor-

tunities created by the web for new technologies; others, such as the thin-client systems proposed in the late 1990s, are gone before they almost arrive. While it isn't necessarily obvious which category a new trend belongs in, a topic should not be investigated unless you are confident that it will continue to be relevant.

Another important question is, is this project at the right kind of technical level? Some brilliant students are neither fast programmers nor systems experts, while others do not have strong mathematical ability. It is not wise to select a project for which you do not have the skills or that doesn't make use of your particular strengths. An alternative perspective on this question is that most projects that are intellectually challenging are interesting to undertake; agonizing over whether it is *the* project may not be productive. However, it is also true that some researchers only enjoy their work if they can identify a broader value: for example, they can see likely practical outcomes. Highly speculative projects leave some people dissatisfied, while others are excited by a possible leap into the future.

Project scope is a related issue. Students can be wildly ambitious, entering research with the hope of achieving something of dramatic significance. However, major breakthroughs are by definition rare—otherwise they wouldn't be major—while, as most researchers discover, even incremental work can be profoundly rewarding. Moreover, an ambitious project creates a high potential for failure, especially in a limited-term context such as a minor thesis. There is a piece of folklore that says that most scientists do their best work in their PhD, as it is the one opportunity to undertake a lengthy, focused research program. This is a myth, and is certainly not a good reason for tackling a problem that is too large to resolve.

Most research is incremental: improvements or variants that improve or repair or extend or replace work done by others. The issue is the scope of the increment. A trivial step that does no more than explore the obvious solution to a simple problem—a change, say, to the fields in a network packet to save a couple of bits—is not worth investigating. There needs to be challenge and the possibility of unexpected discovery for research to be interesting.

For a novice researcher, it makes sense to identify easily achieved outcomes; this is research training, after all, not research olympics. If these outcomes are reached, in a well-designed project it should be easy to move on to more challenging goals.

Some research concerns problems that appear to be solved in commercial software. Often, however, research on such problems is not hard to justify. In a typical implementation the task is to find a workable solution, while in research the task is to measure the quality of the solution, and thus work on the

same problem that produces similar solutions can nonetheless have different outcomes. Moreover, while it is in a company's interests to claim that a problem is solved by their technology, such claims are not easily verified. From this perspective, investigation of a problem for which there is already a commercial solution can be of more value than investigation of a problem of purely academic interest.

Students and advisors

The role of an advisor is a rich one. There are said to be as many scientific methods as there are scientists, but there are more advising styles than there are advisors. Every student–advisor relationship is different.

Advisors are powerful figures in their students' lives. Some professors at the peak of their careers still have strong views—often outrage or amazement—about their own advisors, despite many years of experience on the other side of the fence. Tales include that of the student who saw his advisor twice, once to choose a topic and once to submit; and that of the advisor who casually advised a student to "have another look at some of those famous open problems". Thankfully these are rare exceptions, and are even less acceptable today than they were a decade or two ago.

The purpose of a research program—a PhD, masters, or minor thesis—is for the university to provide a student with research training, while the student demonstrates the capability to undertake research from conception to write-up. A side-benefit is that the student, often with the advisor, should produce some publishable research. There are a range of approaches to advising that achieve these aims, but they are all based on the strategy of learning while doing.

Some advisors, for example, set their students problems such as verifying a proof in a published paper and seeing whether it can be applied to variants of the theorem, thus beginning to explore the limits at which the theorem no longer applies. Another example is to attempt to confirm someone else's results, by downloading code or by developing a fresh implementation. The difficulties encountered in such efforts are a fertile source of research questions. Other advisors immediately start their students on activities that are expected to lead to a research publication. It is in such cases that the model of advising as apprenticeship is most evident.

Typically, in the early stages the advisor specifies each small step the student should take: running a certain experiment, searching the literature to resolve a particular question, or writing one small section of a proposed paper. As students mature as researchers, they become more independent, often by

anticipating what their advisors will ask, while advisors gradually leave more space for their students to assert this independence. Over time, the relationship becomes one of guidance rather than management.

The trade-offs implicit in such a relationship are complex. One is the question of authorship of work the student has undertaken, as discussed in Chapter 13. Another is the degree of independence. Advisors often believe that their students are either demanding or overconfident; students, on the other hand, can feel either confined by excessive control or at a loss due to being expected to undertake tasks without guidance. The needs of students who are working more or less alone may be very different to those of students who are part of an extended research group.

An area where the advisor's expertise is critical is in scoping the project. It needs to stand sufficiently alone from other current work, yet be relevant to a group's wider activities. It should be open enough to allow innovation and freedom, yet have a good likelihood of success. It should be close enough to the advisor's core expertise to allow the advisor to verify that the work is sufficiently novel, and to verify that the appropriate literature has been thoroughly explored. The fact that an advisor finds a topic interesting does not by itself justify asking a student to work on it. Likewise, a student who is keen on a topic must consider whether competent supervision is available in that area.

Advisors can be busy people. Prepare for your meetings—bring printouts of results or lists of questions, for example. Be honest; if you are trying to convince your advisor that you have completed some particular piece of work, then the work should have been done. Advisors are not fools. Saying that you have been reading for a week sounds like an excuse; and, if it is true, you probably haven't spent your time effectively.

The student–advisor relationship is not only concerned with research training, but is a means for advisors to be involved in research on a particular topic. Thus students and advisors often write papers together. At times, this can be a source of conflict, when, for example, an advisor wants a student to work on a paper while the student wants to make progress on a thesis. On the other hand, the involvement of the advisor—and the incentive for the advisor to take an active role—means that the research is undertaken as teamwork.

Finding research literature

Each research project builds on a body of prior work. The doing and describing of research requires a thorough knowledge of the work of others.

However, locating prior work can be a tremendous challenge. The number

of papers published in computer science each year is at least tens of thousands. Not only is a great deal of this work relatively inaccessible, but the volume of it prohibits reading or understanding more than a fraction of the papers appearing in any one field.

A consolation is that, in an active field, other researchers have to a certain extent already explored and digested the older literature. Their work provides a guide to earlier research—as will your work, once it is published—and thus a complete exploration of the archives is rarely necessary. However, this is one more reason to carefully explore current work. And note: reading about a paper that seems relevant is *never* a substitute for reading the paper itself. If you need to discuss or cite a paper, read it first.

Comprehensively exploring relevant literature involves following several intertwined paths.

- Visit the web sites of research groups and researchers working in the area. The web site of your advisor or department is likely to be a good place to start. These sites should give several kinds of links into the wider literature: the names of researchers whose work you should investigate, the names of their co-authors, conferences where relevant work appears, and papers with lists of references to explore.
- Follow up references in research papers. These indicate relevant individuals, conferences, and journals.
- Browse the recent issues of the journals and conferences in the area; search other journals and conferences that might carry relevant papers.
- Use obvious search terms to explore the web. With the right terms you are likely to find the sites of projects and teams concerned with the same research area. You are also likely to find documents that suggest further valuable search terms.
- Search the publisher-specific digital libraries. These include publishers such as Wiley and Springer, and professional societies such as the ACM and IEEE. There are also a wide range of online archives and abstract-indexing services.
- Most conferences have web sites that list the program, that is, the papers to appear in the conference that year. Within a conference, papers are often grouped by topic—another hint of relevance.
- Use the citation indexes. The traditional printed citation indexes have migrated to the web, but in practice their value for computer science is limited, as only a fraction of publications are included. Of much greater value

are the public-domain indexes, which can be used to search by document content and by citing or cited document. Some of these are constructed automatically; others are built by contribution from users. Thus their contents are unreliable, and the origin of documents found in these indexes should be verified elsewhere.

- Go to the library. The simple strategy of having similar material shelved together often leads to unexpected discoveries, without the distractions that arise when web browsing.
- Discuss your work with as many people as possible. Some of them may well know of relevant work you haven't encountered. Similar problems often arise in disparate research areas, but the difficulties of keeping up with other fields—the phenomenon sometimes characterized as "working in silos"—mean that people investigating similar problems can be unaware of one another.

Take a broad definition of "relevant" when searching for papers. It doesn't just mean those papers that have, say, proposals for competing methods. Does the paper have interesting insights into other research literature? Does it establish a benchmark? Have the authors found a clever way of proving a theorem that you can apply in your own work? Does the paper justify not pursuing some particular line of investigation? Other people's research can have many different kinds of effect on your work.

Reading

A thorough search of the literature can easily lead to discovery of dozens or hundred of relevant papers—a volume of reading that can be deeply intimidating. However, papers are not textbooks, and should not be treated as textbooks. A researcher reading a paper is not cramming for an exam; there is rarely a need to understand every line. The number of papers that a researcher working on a particular project has to know well is usually small, even though the number the researcher should have read to establish their relevance is large. A brief browse through a paper takes no more than a few minutes, if the aim is to identify whether the paper is relevant to a particular project.

A problem with dredging the web for research literature is whether to believe what you read. Work published in a reputable journal or conference is peer-reviewed; work available online could have any history, from being a pre-publication version of an accepted journal paper to plagiarised work taken from

a non-English original and rejected from three conferences. A cynical but often accurate rule of thumb is that work that is more than one or two years old and has not been published in a reputable venue probably has some serious defect. When you find a version of a paper on the web, establish whether it has been published somewhere. Use evidence such as the quality of the authors' other publications to establish whether it is part of a serious program of research.

Much research—far too much—is just misguided. People investigate problems that are already solved and well understood, or solve problems that technology has made irrelevant, or try to square the circle (such as attempting to adjust optimal codes to achieve better compression), or don't realise that the proposed improvement actually makes the algorithm worse. Mathematics may be pointless; the wrong property may be proved, such as complexity instead of correctness; assumptions may be implausible; evaluation strategies may not make sense. The data set used may be so tiny that the results are meaningless; results on toy problems rarely scale up. Some results are just plain wrong.

And, while the fact that a paper is refereed is an indicator that it is of value, it is not a guarantee. Too many people submit work that did not deserve to be written; sometimes it gets published.

Indeed, few papers are perfect. They are a presentation of new work rather than a considered explanation of well-known results, and the constraints of writing to a deadline mean that mistakes are undiscovered and some issues unexplored. Some aspects of the work may be superseded or irrelevant, or may rely on false or limited or technically outdated assumptions. A paper can be seen as a snapshot of a research program at a moment in time—what the researchers knew when they submitted. For all these reasons, a reader needs to be questioning and skeptical.

But that does not justify researchers being dismissive of past work; rather, they should respect it and learn from it, because their own work will have the same strengths and weaknesses. While many papers may be flawed, they are the repository of all scientific knowledge—they define scientific knowledge. (Textbooks are almost invariably consolidations of older, established work that is no longer at the frontier.) Moreover, a general view that some papers are unreliable is a poor reason to neglect a particular paper with which you happen to disagree; it may contain an unpalatable truth. And this general view is an extremely poor reason to curtail either your reading or your attempts to understand the contributions made by others. If many researchers trust a particular paper, it is still reasonable to be skeptical of its results, but this needs to be balanced against the fact that, if skepticism is justified, these other researchers are all mistaken.

Read papers by asking questions of them, such as:

- What is the main result?
- How precise are the claims?
- How could the outcomes be used?
- What is the evidence?
- How was the evidence gathered?
- How were measurements taken?
- How carefully are the algorithms and experiments described?
- Why is the paper trustworthy?
- Has the right background literature been discussed?
- What would reproduction of the results involve?

That is, actively attempt to identify the contributions and shortcomings rather than simply reading from one end to the other. Detailed analysis can be difficult before you have developed the perspective of undertaking your own work, however. Literature review should continue alongside research, not precede it.

Capture information about each paper you expect to cite, or of even peripheral relevance. Many of the online services link to a `bibtex` citation; take a copy and annotate it with your own views on the paper. Classify the paper, and cross-index it with others on the same topic. Be organized with such material from the start—don't expect to have time to reinvestigate the literature in detail when completion date is looming.

Having explored the literature, you may discover that your original idea is not so original after all. If so, be honest—review your work to see what aspects may be novel, but don't fool yourself into working on a problem that is already solved. Occasionally it happens, for example, that the same problem has been investigated by several other teams over a considerable period. At the same time, the fact that other people have worked on the same problem does not mean that it is impossible to make further contributions in the area.

Research planning

Students commencing their first research project are accustomed to the patterns of undergraduate study: attending lectures, completing assignments, revising for exams. Activity is determined by a succession of deadlines that impose a great deal of structure.

In contrast, a typical research project has just one deadline: completion. Administrative requirements may impose some additional milestones, such as submission of a project outline or a progress report, but many students (and advisors) do not take these milestones seriously. However, having a series of deadlines is critical to the success of a project. The question then is, what should these deadlines be and how should they be determined?

Some people appear to plan their projects directly in terms of the aspects of the problem that attracted them in the first place. For example, they download some code or implement something, then experiment, then write up. A common failing that seems to arise with this approach to research is that each stage takes longer than anticipated, the time for write-up is compressed, and the final report is compromised. Yet the write-up is the only part of the work that survives or is assessed. Arguably, an even more significant failing is that the scientific validity of the outcomes can be compromised. It is a mistake, for example, to implement a complete system rather than ask what code is needed to explore the research questions.

A better approach to the task of scoping a project and setting milestones is to explicitly consider what is needed at the end, then reason backwards. The final thing required is the write-up in the form of a thesis, paper, or report; so plan in terms of the steps necessary to produce the write-up. Considering as an example research that is expected to have a substantial experimental component, the write-up is likely to involve a background review, explanations of previous and new algorithms, descriptions of experiments, and analysis of outcomes. Completion of each of these elements is a milestone.

Continuing to reason backwards, the next step is to identify what form the experiments will take. Chapter 11 concerns experiments and how they are reported, but prior to designing experiments the researcher must consider how they are to be used. What will the experiments show, assuming the hypothesis to be true? How will the results be different if the hypothesis is false? That is, the experiments are an evaluation of whether some hypothesised phenomena is actually observed. Experiments involve data, code, and some kind of platform. Running of experiments requires that all three of these be obtained, and that skeptical questions be asked about them: whether the data is realistic, for example.

Experiments may also involve users. Who will they be? Is ethics clearance required? Computer scientists, accustomed to working with algorithms and proofs, are often surprised by how wide-ranging their university ethics requirements can be.

Considering work that is not expected to have an experimental component,

there are two general kinds: formal investigations of the properties of systems and algorithms; and a wide range of studies that are difficult to classify, from proposals for new programming language features and sketches of XML templates for particular kinds of data to reflections on and comparisons of trends in research. Each of these can be staged to identify research milestones.

Drawing these issues together, several themes emerge. One is that the components of research have to be identified; however, these components do not necessarily have to be completed in turn.

Another theme is that an attitude to research has been shaped: what information must be collected in order to convince a skeptical reader that the results are correct? Arguably, answering this one question is all that is needed to have a strong research outcome.

Having identified specific goals, another purpose of research planning is to estimate dates when milestones should be reached. One of the axioms of research, however, is that everything takes longer than planned for, even after taking this axiom into account. A standard research strategy is to first read the literature, then design, then analyse or implement, then test or evaluate, then write up. A more effective strategy is to overlap these stages as much as possible. Begin the implementation, or analysis, or write up as soon as it is reasonable to do so.

For the longer-term research of, for example, a PhD, other considerations become significant. A typical question in the later stages of a PhD is whether enough research has yet been done, or whether new additional work needs to be undertaken. Often the best response to this question is to write the thesis. Once your thesis is more or less complete, it is relatively easy to assess whether further work is justified. Doing such additional work in all likelihood involves filling a well-defined hole, a task that is much better defined than that of fumbling around for further questions to investigate.

Thus, rather than working to a schedule of long-term timelines that may be unrealistic, be flexible. Adjust the work you are doing on a day-to-day basis—pruning your research goals, giving more time to the writing, addressing whatever the current bottleneck happens to be—to ensure that you are reaching overall aims.

Hypotheses

The first stages of a research program involve identifying interesting topics or problems and focusing on particular issues to investigate. A typical way of giving direction to research is to develop specific questions that the program

aims to answer. These questions are based on an understanding, an informal model perhaps, of how something works, or interacts, or behaves. They establish a framework for making observations about the object being studied. This framework can be characterised as a statement of belief about how the object behaves—in other words, a hypothesis.

In the traditional sciences, a hypothesis typically concerns some phenomenon in the physical world: whether something is occurring, or whether it is possible to alter something in a predictable way. Astronomy and genetics typify such research. In computer science, some hypotheses are of this kind. Other hypotheses involve construction, such as whether a proposed method is fit for a certain purpose, and solvability.

For example, a researcher investigating algorithms might ask as a *research question* whether it is possible to make better use of CPU cache to reduce computational costs; reducing the number of memory accesses can make a program faster even if the number of instructions executed is unchanged. Preliminary investigation might lead to the *hypothesis* that a particular sorting algorithm can be improved by replacing a tree-based structure with poor locality by an array-based structure with high locality. The *research goal* is to test this hypothesis. The *phenomenon* that should be observed if the hypothesis is correct is a trend: as the number of items to be sorted is increased, the tree-based method should increasingly show a high rate of cache misses compared to the array-based method. The *data* is the number of cache misses for several sets of items to be sorted.

A hypothesis should be specified clearly and precisely, and should be unambiguous. (The more loosely a concept is defined, the more easily it will satisfy many needs simultaneously, even when these are contradictory.) Often it is important to state what is *not* being proposed—what the limits on the conclusions will be. Consider an example. Suppose P-lists are a well-known data structure used for a range of applications, in particular as an in-memory search structure that is fast and compact. A scientist has developed a new data structure called the Q-list. Formal analysis has shown the two structures to have the same asymptotic complexity in both space and time, but the scientist intuitively believes the Q-list to be superior in practice and has decided to demonstrate this by experiment.

(This motivation by belief, or instinct, is a crucial element of the process of science: since ideas cannot be known to be correct when first conceived, it is intuition or plausibility that suggests them as worthy of consideration. That is, the investigation may well have been undertaken for subjective reasons; but the final report on the research, the published paper, must be objective.)

The hypothesis might be encapsulated as

✗ Q-lists are superior to P-lists.

But this statement does not suffice as the basis of experiment: success would have to apply in all applications, in all conditions, for all time. Formal analysis might be able to justify such a result, but no experiment will be so far-reaching. In any case, it is rare indeed for a data structure to be completely superseded—consider the durability of arrays and linked lists—so in all probability this hypothesis is incorrect. A testable hypothesis might be

✓ As an in-memory search structure for large data sets, Q-lists are faster and more compact than P-lists.

Further qualification may well be necessary.

✓ We assume there is a skew access pattern, that is, that the majority of accesses will be to a small proportion of the data.

The qualifying statement imposes a scope on the claims made on behalf of Q-lists. A reader of the hypothesis has enough information to reasonably conclude that Q-lists do not suit a certain application, which in no way invalidates the result. Another scientist would be free to explore the behaviour of Q-lists under another set of conditions, in which they might be inferior to P-lists, but again the original result remains valid.

As the example illustrates, a hypothesis must be testable. One aspect of testability is that the scope be limited to a domain that can feasibly be explored. Another, crucial aspect is that the hypothesis should be capable of falsification. Vague claims are unlikely to meet this criterion.

✗ Q-list performance is comparable to P-list performance.

✗ Our proposed query language is relatively easy to learn.

The exercise of refining and clarifying a hypothesis may expose that it is not worth pursuing. For example, if complex restrictions must be imposed to make the hypothesis work, or if it is necessary to assume that current insoluble problems need to be addressed before the work can be used, how interesting is the research?

A form of research where poor hypotheses seem particularly common is "black box" work, where the black box is an algorithm whose properties are

incompletely understood. For example, some research consists of applying a black-box learning algorithm to new data, with the outcome that the results are an improvement on a baseline method. (Often, the claim is to the effect that "our black box is significantly better than random".) The apparent ability of these black boxes to solve problems without creative input from a scientist attracts research of low value. A weakness of such research is that it provides no learning about the data or the black box, and has no implications for other investigations. In particular, such results rarely tell us whether the same behaviour would occur the next time the same approach was used.

A related problem is the renaming fallacy, often observed in the work of scientists who are attempting to reposition their research within a fashionable area. Calling a network cache a "local storage agent" doesn't change its behaviour, and if the term "agent" can legitimately be applied to any executable process then its explanatory power is slim. Another instance: a paper on natural language processing for "web documents" should concern some issues specific to the web, not just any text; a debatable applicability to the web does not add to the contribution. And another: it seems unlikely that a text indexing algorithm is made "intelligent" by improvements to the parsing. Renaming existing research to place it in another field is bad science.

It may be necessary to refine a hypothesis as a result of initial testing; indeed, much of scientific progress can be viewed as refinement and development of hypotheses to fit new observations. Occasionally there is no room for refinement, a classic example being Einstein's prediction of the deflection of light by massive bodies—a hypothesis much exposed to disproof, since it was believed that significant deviation from the predicted value would invalidate the theory of general relativity. But more typically a hypothesis evolves in tandem with refinements in the experiments.

This is not, however, to say that the hypothesis should follow the experiments. A hypothesis will often be based on observations, but can only be regarded as confirmed if able to make successful predictions. There is a vast difference between an observation such as "the algorithm worked on our data" and a tested hypothesis such as "the algorithm was predicted to work on any data of this class, and this prediction has been confirmed on our data". Another way of regarding this issue is that, as far as possible, tests should be blind. If an experiment and hypothesis have been fine-tuned on the data, it cannot be said that the experiment provides confirmation. At best the experiment has provided observations on which the hypothesis is based.

Where two hypotheses fit the observations equally well and one is clearly simpler than the other, the simpler should be chosen. This principle, known as

Occam's razor, is purely a convenience; but it is well-established and there is certainly no reason to choose a complex explanation when another is at hand.

Defending hypotheses

One component of a strong paper is a precise, interesting hypothesis. Another component is the testing of the hypothesis and the presentation of the supporting evidence. As part of the research process you need to test your hypothesis and if it is correct—or, at least, not falsified—assemble supporting evidence. For the presentation of the hypothesis you need to construct an argument relating your hypothesis to the evidence.

For example, the hypothesis "the new range searching method is faster than previous methods" might be supported by the evidence "range search amongst n elements requires $2\log_2\log_2 n + c$ comparisons". This may or may not be good evidence, but it is not convincing because there is no argument connecting the evidence to the hypothesis. What is missing is information such as "previous results indicated a complexity of $\Theta(\log n)$". It is the role of the connecting argument to show that the evidence does indeed support the hypothesis, and to show that conclusions have been drawn correctly.

In constructing an argument, it can be helpful to imagine yourself defending your hypothesis to a colleague, so that you play the role of inquisitor. That is, raising objections and defending yourself against them is a way of gathering the material needed to convince the reader that your argument is correct. Starting from the hypothesis that "the new string hashing algorithm is fast because it doesn't use multiplication or division" you might debate as follows:

- I don't see why multiplication and division are a problem.

  On most machines they use several cycles, or may not be implemented in hardware at all. The new algorithm instead uses two exclusive-or operations per character and a modulo in the final step. I agree that for pipelined machines with floating-point accelerators the difference may not be great.

- Modulo isn't always in hardware either.

  True, but it is only required once.

- So there is also an array lookup? That can be slow.

  Not if the array is cache-resident.

- What happens if the hash table size is not 2^8?

Good point. This function is most effective for hash tables of size 2^8, 2^{16}, and so on.

In an argument you need to rebut likely objections while conceding points that can't be rebutted and admitting when you are uncertain. If, in the process of developing your hypothesis, you raised an objection but reasoned it away, it can be valuable to include the reasoning in the paper. Doing so helps the reader to follow your train of thought, and certainly helps the reader who independently raises the same objection. That is, you need to anticipate problems the reader may have with your hypothesis. Likewise, you should actively search for counter-examples.

If you think of an objection that you cannot refute, don't just put it aside. At the very least you should raise it yourself in the paper, but it may well mean that you must reconsider your results.

A hypothesis can be tested in a preliminary way by considering its effect, that is, by examining whether there is a simple argument for keeping or discarding it. For example, are there any improbable consequences if the hypothesis is true? If so, there is a good chance that the hypothesis is wrong. For a hypothesis that displaces or contradicts some currently held belief, is the contradiction such that the belief can only have been held out of stupidity? Again, the hypothesis is probably wrong. Does the hypothesis cover all of the observations explained by the current belief? If not, the hypothesis is probably uninteresting.

Always consider the possibility that your hypothesis is wrong. It is often the case that a correct hypothesis at times seems dubious—perhaps initially, before it is fully developed, or when it appears to be contradicted by some experimental evidence—but the hypothesis survives and is even strengthened by test and refinement in the face of doubt. But equally often a hypothesis is false, in which case clinging to it is a waste of time. Persist for long enough to establish whether or not it is likely to be true, but to persist longer is foolish.

A corollary is that the stronger your intuitive liking for a hypothesis, the more rigorously you should test it—attempt to confirm it or disprove it—rather than twist results, and yourself, defending it.

Be persuasive. Using research into the properties of an algorithm as an example, issues such as the following need to be addressed.

- Will the reader believe that the algorithm is new?

  Only if the researcher does a careful literature review, and fully explores and explains previous relevant work. Doing so includes giving credit to significant advances, and not overrating work where the contribution is small.

- Will the reader believe that the algorithm is sensible?

  It had better be explained carefully. Potential problems should be identified, and either conceded—with an explanation, for example, of why the algorithm is not universally applicable—or dismissed through some cogent argument.

- Are the experiments convincing?

  If the code isn't good enough to be made publicly available, is it because there is something wrong with it? Has the right data been used? Has enough data been used?

Every research program suggests its own questions. Such questioning is also appropriate later in a research program, where it provides an opportunity for critical assessment of the work.

Evidence

A view of papers is that they are an assembly of evidence and supporting explanation, that is, an attempt to persuade others to share your conclusions. In a write-up you pose a hypothesis, then present evidence to support your case. The evidence needs to be convincing because the processes of science rely on readers being critical and skeptical; there is no reason for a reader to be interested in work that is inconclusive. There are, broadly speaking, four kinds of evidence that can be used to support a hypothesis: analysis or proof, modelling, simulation, and experiment.

An analysis or proof is a formal argument that the hypothesis is correct. It is a mistake to suppose that the correctness of a proof is absolute—confidence in a proof may be high, but that does not guarantee that it is free from error. (In my experience it is not uncommon for a researcher to feel certain that a theorem is correct but have doubts about the mechanics of the proof, which all too often leads to the discovery that the theorem is wrong after all.) And it is a mistake to suppose that all hypotheses are amenable to formal analysis, particularly hypotheses that involve the real world in some way. For example, human behaviour is intrinsic to questions about interface design, and system properties can be intractably complex. Consider an exploration to determine whether a new method is better than a previous one at compressing text—is it likely that something as diverse as text can be modelled well enough to predict the performance of a compression algorithm? It is also a mistake to suppose

that a complexity analysis is always sufficient. Nonetheless, the possibility of formal analysis should never be overlooked.

A model is a mathematical description of the hypothesis (or some component of the hypothesis such as an algorithm whose properties are being considered) and there will usually be a demonstration that the hypothesis and model do indeed correspond.

In choosing to use a model, consider how realistic it will be, or conversely how many simplifying assumptions need to be made for analysis to be feasible. Consider the example of modelling the cost of a Boolean query on a text collection, in which the task is to find the documents that contain each of a set of words. We need to estimate the frequency of each word (because words that are frequent in queries may be rare in documents); the likelihood of query terms occurring in the same document (in practice, query terms are thematically related, and do not model well as random co-occurrences); the fact that longer documents contain more words, but are more expensive to fetch; and, in a practical system, the probability that the same query had been issued recently and the answers are cached in memory. It is possible to define a model based on these factors, but, with so much guesswork to make, it is unlikely that the model would be realistic.

A simulation is usually an implementation or partial implementation of a simplified form of the hypothesis, in which the difficulties of a full implementation are sidestepped by omission or approximation. At one extreme a simulation might be skeletal, so that, for example, a parallel algorithm could be tested on a sequential machine by use of an interpreter that counts machine cycles and communication costs between simulated processors; at the other extreme a simulation could be an implementation of the hypothesis, but tested by artificial data. A simulation is a "white coats" test: artificial, isolated, and conducted in a tightly controlled environment.

An experiment is a full test of the hypothesis, based on an implementation of the proposal and on real—or at least realistic—data. In an experiment there is a sense of *really doing it*, while in a simulation there is a sense of *only pretending*. However, the distinction between simulation and experiment can be blurry.

Ideally an experiment should be conducted in the light of predictions made by a model, so that it confirms some expected behaviour. An experiment should be severe; look for tests that are likely to fail if the hypothesis is false. The traditional sciences, and physics in particular, proceed in this way. Theoreticians develop models of phenomena that fit known observations; experimentalists seek confirmation through fresh experiments.

Different forms of evidence can be used to confirm one another, with say a simulation used to provide further evidence that a proof is correct. But they should not be confused with one another. For example, suppose that for some algorithm there is a mathematical model of expected performance. Encoding this model in a program and computing predicted performance for certain values of the model parameters is in no way an experimental test of the algorithm and should never be called an experiment; it does not even confirm that the model is a description of the algorithm. At best it confirms claimed properties of the model.

When choosing whether to use a proof, model, simulation, or experiment as evidence, consider how convincing each is likely to be to the reader. If your evidence is questionable—say a model based on simplifications and assumptions, an involved algebraic analysis and application of advanced statistics, or an experiment on limited data—the reader may well be skeptical of the result. Select a form of evidence, not so as to keep your own effort to a minimum, but to be as persuasive as possible.

Having identified the elements a research plan should cover, end-to-start reasoning also suggests priorities. The write-up is the most important thing; so perhaps it should be started first. Completing the report is certainly more important than hastily running some last-minute experiments, or quickly browsing the literature to make it appear as if past work has been fully evaluated.

Good and bad science

Questions about the quality of evidence can be used to evaluate other people's research, and provide an opportunity to reflect on whether the outcomes of your work are worthwhile. There isn't a simple division of research into "good" and "bad", but it is not difficult to distinguish valuable research from work that is weak or pointless.

The merits of formal studies are easy to appreciate. They provide the kind of mathematical link between the possible and the practical that physics provides between the universe and engineering.

The merits of well-designed experimental work are also clear. Work that experimentally confirms or contradicts the correctness of formal studies has historically been undervalued in computer science: perhaps because standards for experimentation have not been high; perhaps because the great diversity of computer systems, languages, and data has made truly general experiments difficult to devise; perhaps because theoretical work with advanced mathematics is more intellectually imposing than work that some people regard as mere

code-cutting. However, many questions cannot be readily answered through analysis, and a theory without practical confirmation is no more interesting in computing than in the rest of science.

However, research that consists of proposals—without a serious attempt at evaluation—can be more difficult to respect. Why should a reader regard such work as valid? If the author cannot offer anything to measure, arguably it isn't science. As discussed in Chapter 11, there are many ways of measuring a system or result. And research isn't theoretical just because it isn't experimental. Theoretical work describes testable theories.

Some science is not simply weak, but can be classed as pseudoscience. A great deal of money can be made by appearing to have solved major problems, and scientists seek prestige through their research achievements. Inevitably, some claimed achievements are delusional or bogus.

Pseudoscience is a broad label covering a range of scientific sins, from self-deception and confusion to outright fraud. A definition is that pseudoscience is work that uses the language and respectability of science to gain credibility for statements that are not based on evidence that meets scientific standards. Much pseudoscience shares a range of characteristics: the results and ideas don't seem to develop over time, systems are never quite ready for demonstration, the work proceeds in a vacuum and is unaffected by other advances, protagonists argue rather than seek evidence, and the results are inconsistent with accepted facts. Often such work is strenuously promoted by one individual or a small number of devotees while the rest of the scientific community ignores it.

An example of pseudoscience in commercial computing is some of the schemes for high-performance video compression, which promise delivery of TV-quality data over 56 kilobaud modems. The commercial implications of such systems are enormous, and this incentive creates ample opportunities for fraud; in one case, for example, millions of dollars were scammed from investors with tricks such as hiding a video player inside a PC tower and hiding a network cable inside a power cable. Yet, skeptically considered, such schemes are implausible. For example, with current technology, even a corner of a single TV-resolution image—let alone 25 frames per second—cannot be compressed into 7 kilobytes. Uncompressed, the bandwidth of a modem is only sufficient for one byte per row per image, or, per image, about the space needed to transmit a desktop icon. A further skeptical consideration in this case was that an audio signal was also transmitted. Had the system been legitimate, the inventor must have solved the independent problems of image compression, motion encoding, and audio compression.

It is not hard to find similar work in the academic world. An example

is the variety of "universal" indexing methods that have been proposed. In these methods, the object to be indexed—whether an image, movie, audio file, or text document—is manipulated in some way, for example by a particular kind of hash function. After this manipulation, objects of different type can be compared: thus, somehow, documents about swimming pools and images of swimming pools would have the same representation. Such matching is clearly an extremely difficult problem, if not entirely insoluble; for instance, how does the method know to focus on the swimming pool rather than some other element of the image, such as children, sunshine, or a metaphor for middle class aspirations? Yet proposals for such methods continue to appear. In a recent version, objects of the same type were clustered together using some kind of similarity metric. Then the patterns of clustering were analysed, and objects that clustered in similar ways were supposed to have similar subject matter. Although it is disguised by the use of clustering, to be successful such an approach assumes an underlying universal matching method.

In some work, the evidence or methods are inconsistent. For example, in a paper on how to find documents on a particular topic, the authors reported that the method correctly identified 20,000 matches in a large document collection. But this is a deeply improbable outcome. The figure of 20,000 hints at imprecision—it is too round a number. More significantly, verifying that all 20,000 were matches would require many months of effort. No mention was made of the the documents that weren't matches, implying that the method was 100% accurate; but even the best document-matching methods have high error rates. A later paper by the same authors gave entirely different results for the same method, while claiming similar good results for a new method, thus throwing doubt on the whole research program. And it is a failure of logic to suppose that the fact that two documents match according to some arbitrary algorithm implies that the match is useful to a user.

The logic underlying some papers is downright mystifying. It may seem a major step to identify and solve a new problem, but such steps can go too far. A paper on retrieval for a specific form of graph used a new query language and matching technique, a new way of evaluating similarity, and data based on a new technique for deriving the graphs from text and semantically labelling the edges. Every element of this paper was a separate contribution whose merit could be disputed. Presented in a brief paper, the work seemed worthless. Inventing a problem, a solution to the problem, and a measure of the solution—all without external justification—is a widespread form of bad science.

An interesting question is how to regard "Zipf's law". This observation—"law" seems a poor choice of terminology in this context—is if nothing else

a curious case study. Zipf's books may be widely cited but they are not, I suspect, widely read. In *Human Behaviour and the Principle of Least Effort* (Addison-Wesley, 1949), Zipf used languages and word frequencies as one of several examples to illustrate his observation, but his motivation for the work is not quite what might be expected. He states, for example, that his research "define[s] objectively what we mean by the term personality" (p. 18), explains the "drives of the Freudian death wish" (p. 17), and "will provide an objective language in terms of which persons can discuss social problems impersonally" (p. 543). It "will help to protect mankind from the virtual criminal action of persons in strategic political, commercial, social, intellectual and academic positions" (p. 544) and "as the authority of revealed religion and its attendant ethics declines, something must take its place ... I feel that this type of research may yield results that will fulfill those needs" (p. 544). Perhaps these extraordinary claims are quirks, and in any case opinions do not invalidate scientific results. But it has been argued that the behaviour captured by Zipf's conjecture is a simple consequence of randomness, and, for the motivating example for which the conjecture is often cited (distribution of words in text), the fit between hypothesis and observation is not always strong.

A lesson is that we need to be wary of claimed results, not only because we might disagree for technical reasons but because the behaviour of other researchers may not be objective or reasonable. Another lesson is that acceptance of (or silence about) pseudoscience erodes the perceived need for responsible research, and that it is always reasonable to ask skeptical questions. Yet another lesson is that we need to take care to ensure that our own research is well founded. When results are defended by assertion, with no evaluation or evidence, it is easy to wonder whether the work is an instance of pseudoscience.

Reflections on research

Philosophies and definitions of science establish guidelines for what scientists do and set boundaries on what we can know. However, there are limits to how precise (or interesting) such definitions can be. For example, the question "is computer science a science?" has a low information content.[14] Questions of this kind are sometimes in terms of definitions of science such as "a process

[14]Two philosophers are arguing in a bar. The barman goes over to them and asks, "What are you arguing about?"

"We're debating whether computer science is a science", answers one of them.

"And what do you conclude?" asks the barman.

"We're not sure yet," says the other. "We can't agree on what 'is' means."

for discovering laws that model observed natural phenomena". Such definitions not only exclude disciplines such as computing, but also exclude much of the research now undertaken in disciplines such as biology and medicine. In considering definitions of science, a certain degree of skepticism is valuable; these definitions are made by scientists working within particular disciplines and within the viewpoints that those disciplines impose. In fairness, I note that the views below have the same limitations, as they are those of a computer scientist who believes that the discipline stands alongside the traditional sciences.

It is true that, considered as a science, computing is difficult to categorize. The underlying theories—information theory and computability, for example—appear to describe properties as eternal as those of physics. (Such properties can be seen as constraints separating the possible from the impossible.) In recent years the distinction between the laws of computing and the laws of physics has blurred, with for example properties of black holes being described by information theory. Yet most research in computer science is many steps removed from foundational theory and more closely resembles engineering or psychology.

A widely agreed description of science is that it is a method for accumulating reliable knowledge. In this viewpoint, scientists adopt the belief that rationality and skepticism are how we learn about the universe and shape new principles, while recognizing that this belief limits the application of science to those ideas that can be examined in a logical way. If the arguments and experiments are sound, if the theory can withstand skeptical scrutiny, if the work was undertaken within a framework of past research and provides a basis for further discovery, then it is science. Much computer science has this form.

Many writers and philosophers have debated the nature of science, and aspects of it such as the validity of different approaches to reasoning. The direct impact of this debate on the day-to-day activity of scientists is small, but it has undoubtedly shaped how scientists approach their work. It also provides elements of the ethical framework within which scientists work.

A key effect of philosophy on the practice of science has been to undermine belief in certainty and absolute scientific truth. Several developments in the early years of the twentieth century contributed to this development, including relativity, quantum mechanics, incompleteness, and undecidability. In philosophy, the ideal of scientific truth was undermined by the concept of falsification. The core idea is simple: experimental evidence, no matter how substantial or voluminous, cannot prove a theory true, while a single counter-example can prove a theory false.

A practical consequence of the principle of falsification is that a reasonable

scientific method is to search for counter-examples to hypotheses. In this line of reasoning, to search for supporting evidence is pointless, as such evidence cannot tell us that the theory is true. A drawback of this line of reasoning is that, using falsification alone, we cannot learn any new theories; we can only learn that some theories are wrong. Another issue is that, in practice, experiments are often unsuccessful, but the explanation is not that the hypothesis is wrong, but rather that some other assumption was wrong. The response of a scientist to a failed experiment may well be to redesign it. For example, in the decades-long search for gravity waves, there have been many unsuccessful experiments, but a general interpretation of these experiments has been that they show that the equipment is insufficiently sensitive.

Thus falsification can be a valuable guide to the conduct of research, but other guides are also required if the research is to be productive. One such guide is the concept of confirmation. In science, "confirmation" has a weaker meaning than in general usage; when a theory is confirmed, the intended meaning is not that the theory is proved, but that the weight of belief in the theory has been strengthened. Seeking experiments that confirm theories is an alternative reasonable view of method.

A further consequence is that a hypothesis should allow some possibility of being disproved—there should be some experiment whose outcomes could show that they hypothesis is wrong. If not, the hypothesis is simply uninteresting. Consider, for example, the hypothesis "a search engine can find interesting web pages in response to queries". It is difficult to see how this supposition might be contradicted. Thus falsification and other descriptions of method help shape research questions as well as research processes.

A research checklist

- Are the ideas clear and consistent?
- Is the problem worthy of investigation?
- Does the project have appropriate scope?
- What are the specific research questions?
- Is there a hypothesis?
- What would disprove the hypothesis? Does it have any improbable consequences?
- Are the premises sensible?

- Has the work been critically questioned? Have you satisfied yourself that it is sound science?
- How are the outcomes to be evaluated? Why are the chosen methods of evaluation appropriate or reasonable?
- Are the roles of the participants clear? What are your responsibilities? What activities will the others undertake?
- What are the likely weaknesses of your solution?
- Is there a written research plan?
- What forms of evidence are to be used?
- Have milestones, timelines, and deadlines been identified?
- Do the deadlines leave enough time for your advisor to provide feedback on your drafts, or for your colleagues to contribute to the material?
- Has the literature been explored in appropriate depth? Once the work is largely done—and your perspective has changed—does it need to be explored again?